

第03期-使用命名空间整理类型

2021年12月16日 13:15

C# 程序使用命名空间进行组织。命名空间同时用作程序的 "内部" 组织系统和 "外部" 组织系统，这是一种用于提供向其他程序公开的程序元素的方式。

使用 `using` 指令 (使用指令) 提供以简化命名空间的使用

声明命名空间

使用嵌套关系定义命名空间。

```
namespace N1
{
    namespace N2
    {
        class A {}

        class B {}
    }
}

namespace N1.N2
{
    class A {}

    class B {}
}
```

C# 10 为该文件中定义的所有类型声明一个命名空间

```
namespace SampleNamespace;

class AnotherSampleClass
{
    public void AnotherSampleMethod()
    {
        System.Console.WriteLine("SampleMethod inside SampleNamespace");
    }
}
```

命名空间别名

```
using Project = PC.MyCompany.Project;
```

C# 6.0 引入 static 修饰符导入指定的类型

```
using static <fully-qualified-type-name>;
```

C# 10 引入 global 修饰符全局引入命名空间

```
global using <fully-qualified-namespace>;
```

```
<Using Include="My.Awesome.Namespace" />
```

多个修饰符的命名空间

```
global using static System.Math;
```

命名空间别名限定符

命名空间别名限定符 可以 :: 保证类型名称查找不受新类型和成员引入的影响。命名空间别名限定符始终出现在两个标识符（称为左侧和右侧标识符）之间。与常规 . 限定符不同，限定符的左侧标识符 :: 仅作为 extern 或 using 别名进行查找。

global 命名空间是“根”命名空间：**global::System** 始终引用 .NET System 命名空间。

```
namespace N
{
    public class A {}

    public class B {}
}

namespace N
{
    using A = System.IO;

    class X
    {
        A.Stream s1;           // Error, A is ambiguous

        A::Stream s2;         // Ok
    }
}
```

```
}  
}
```

使用 msbuild 引入命名空间

```
<ImplicitUsings>enable</ImplicitUsings>  
  
<ItemGroup>  
  <Using Include="My.Awesome.Namespace" />  
</ItemGroup>  
  
<ItemGroup>  
  <Using Include="My.Awesome.Namespace" Alias="Awesome" />  
</ItemGroup>  
  
<ItemGroup>  
  <Using Include="Microsoft.AspNetCore.Http.Results" Static="True" />  
</ItemGroup>
```

微软命名空间命名准则

<Company>.(<Product>|<Technology>)[.<Feature>][.<Subnamespace>]

<https://docs.microsoft.com/zh-cn/dotnet/standard/design-guidelines/names-of-namespaces>

<https://docs.microsoft.com/zh-cn/dotnet/standard/design-guidelines/names-of-assemblies-and-dlls>

如何查找指定类型所在的程序集和命名空间

<https://docs.microsoft.com/zh-cn/dotnet/api>

参考资料

<https://rehansaeed.com/the-problem-with-csharp-10-implicit-usings>